

UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Kedua
Sidang Akademik 1996/1997

April 1997

CSM101 - Prinsip-Prinsip Pengaturcaraan

CSC101 - Prinsip-Prinsip Sains Komputer

Masa: [2 jam]

ARAHAN KEPADA CALON:

- Sila pastikan bahawa kertas peperiksaan ini mengandungi **DUA** soalan di dalam **TUJUH** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.
 - Bagi soalan yang melibatkan bahasa pengaturcaraan, bahasa rujukan adalah bahasa pengaturcaraan C.
 - Jawab **SEMUA** soalan dalam Bahasa Malaysia.
-

1. (a) Nyatakan sama ada kenyataan-kenyataan berikut **BENAR** atau **PALSU**.

- (i) Unit kawalan adalah salah satu komponen Unit Pemprosesan Pusat dan fungsi utamanya adalah menyelaras semua fungsi komputer dan mengawal operasi perkakasan.
- (ii) Ingatan Capaian Rawak (RAM) adalah sejenis ingatan kekal yang digunakan untuk menyimpan atur cara dan data pengguna.
- (iii) Terdapat dua jenis data asas yang boleh digunakan dalam pengaturcaraan iaitu **int** dan **char**.
- (iv) Jika salah satu operasi (operand) dalam ungkapan aritmetik adalah berjenis integer, hasil daripada pelaksanaan ungkapan tersebut juga akan berjenis integer.
- (v) Badan bagi gelung **while** mestilah menyebabkan syarat ulangan gelung menjadi palsu untuk menghalang gelung tersebut menjadi tak terhingga.
- (vi) Fungsi matematik **ceil(3.5)** akan memulangkan nilai **3**.
- (vii) Keratan atur cara ini akan memberikan hasil **i** sebagai **5** selepas keratan atur cara dilaksanakan;


```

i = 3;
j = 10;
if ((3 * i) < j)
    i = i + 2;
i = i + 3;
      
```
- (viii) Arahan **c = getch()**, akan membaca satu aksara dari input piawai dan menyalin nilainya ke **c**.
- (ix) Jika sesuatu fungsi yang dipanggil tidak memulangkan nilai (**void**), fungsi tersebut tidak boleh menerima sebarang parameter dari fungsi yang memanggilnya.
- (x) Kenyataan **return** perlu diletak di dalam setiap badan fungsi tanpa mengira sama ada fungsi tersebut akan memulangkan nilai atau tidak.

[30/100]

(b) Kaji algoritma yang diberikan di bawah:

1. Awalkan nilai A kepada 5
2. Ulang dua langkah berikut:
 - 2.1 laporkan A
 - 2.2 tambahkan 5 kepada nilai semasa A
3. Berhenti

- (i) Apakah sifat/keperluan algoritma yang **tidak** dipenuhi oleh algoritma di atas?
- (ii) Betulkan algoritma di atas supaya ia dapat memenuhi sifat/keperluan algoritma.

[15/100]

- (c) Bangunkan satu pseudokod untuk membaca satu siri nombor-nombor positif (satu demi satu) dan tentukan:

- bilangan nombor yang dibaca
- dan nombor terbesar yang dibaca

sehingga nombor 0 dibaca. Seterusnya laporkan semua hasil pengiraan.

[20/100]

- (d) Anda diberi pengisytiharan pemboleh ubah seperti berikut :

```
int      i=8, j=5;
char     c='c', d='d';
```

Apakah yang akan dihasilkan oleh setiap ungkapan aritmetik berikut?

- (i) $(3 * i - 2 * j) \% 2$
- (ii) $++i - j--$
- (iii) $c > d$
- (iv) $(i > 0) \&\& (j < 5)$
- (v) $-(i + j)$

[15/100]

- (e) Diberi satu keratan atur cara seperti berikut:

```
for (i = 1; i <= 6; i++)
{
    if (i % 2)
        continue;
    else
        printf ("%d\n", i);
    printf("bottom of loop\n");
}
```

- (i) Jejak operasi gelung for di atas, dan nyatakan output yang akan dicetak.
- (ii) Tulis kembali keratan atur cara di atas dengan menggunakan gelung while.

[20/100]

2. (a) Huraikan secara ringkas apakah yang dimaksudkan dengan pengaturcaraan bermodul dan senaraikan 3 kebaikannya dalam proses pembangunan atur cara.

[15/100]

- (b) Apakah yang dimaksudkan oleh setiap istilah berikut? Beri contoh dalam bentuk kod atur cara yang sesuai untuk menerangkan tujuan setiap satu dalam proses pembangunan atur cara.

- (i) Definisi fungsi
- (ii) Panggilan fungsi

[20/100]

- (c) Atur cara di bawah menunjukkan satu fungsi **main** memanggil satu fungsi bernama *twice*:

```
#include <stdio.h>
main()
{
    int x=6;
    printf("d\n", twice(x) + 1);
    printf("%d\n", twice(x+1));
}

int twice(int x)
{
    x *=2;
    return (x);
}
```

- (i) Tulis prototaip fungsi untuk *twice*.
- (ii) Berikan output yang akan dicetak oleh atur cara di atas.

[15/100]

- (d) Fungsi faktorial boleh dibangunkan dengan menggunakan rekursi. Definisi fungsi bagi faktorial dikodkan seperti berikut:

```
int faktorial(int n)
{
    return (n * faktorial(n-1));
}
```

Fungsi **faktorial** di atas akan menyebabkan gelung yang tak terhingga disebabkan ketidakwujudan kenyataan kes asas (base case) dalam fungsi tersebut. Tulis kembali definisi fungsi **faktorial** di atas dengan memasukkan kes asas.

[15/100]

(e) Diberi satu masalah seperti berikut:

Mengira gaji kasar pekerja bagi sebuah firma yang mempunyai 100 orang pekerja secara interaktif. Gaji pekerja dibayar setiap minggu bergantung kepada jumlah jam bekerja seminggu dan kadar gaji sejam bagi setiap pekerja di mana gaji kasar = jumlah jam x kadar gaji. Jika jumlah jam bekerja lebih dari 40, jumlah jam yang lebih dari 40 akan menerima kadar gaji $1\frac{1}{2}$ (satu setengah) dari kadar biasa.

Rangka bagi atur cara tersebut adalah seperti di muka surat sebelah (muka surat 6), anda dikehendaki menambahkan kod-kod atur cara di bahagian-bahagian di dalam kotak yang bernombor untuk melengkapkan atur cara tersebut. Setiap komen (/ * teks di dalam ini */) pada baris atas kotak menerangkan secara ringkas apa yang perlu ditambah pada setiap ruang kosong dalam atur cara tersebut.

Spesifikasi bagi setiap fungsi dalam atur cara tersebut adalah seperti berikut:

- Fungsi pertama iaitu **main**. Fungsi ini akan menerima input nombor pekerja, kadar gaji sejam dan jumlah jam bekerja dari pengguna. Jika jumlah jam bekerja ≤ 40 , fungsi ini akan memanggil fungsi kedua bernama **gaji_biasa** dan menghantar nilai kadar gaji sejam dan jumlah jam bekerja kepada fungsi **gaji_biasa**. Jika jumlah jam bekerja > 40 , fungsi ini akan memanggil fungsi ketiga bernama **gaji_lebih_masa** dan menghantar dua nilai yang sama seperti yang dihantar ke fungsi kedua. Fungsi ini kemudian akan mencetak jumlah gaji kasar yang diperolehi oleh pekerja berkenaan.
- Fungsi kedua bernama **gaji_biasa**. Fungsi ini akan menerima dua nilai dari fungsi **main** seperti yang disebutkan di atas dan mengira gaji kasar seminggu. Fungsi ini akan memulangkan jumlah gaji kasar yang diperolehi kepada fungsi **main**.
- Fungsi ketiga bernama **gaji_lebih_masa**. Fungsi ini juga akan menerima dua nilai dari fungsi **main** seperti di dalam fungsi kedua dan mengira gaji beserta gaji lebih masa, Fungsi ini juga akan memulangkan jumlah gaji kasar yang diperolehi kepada fungsi **main**.

(Anda tidak perlu menyalin keseluruhan atur cara ke dalam buku jawapan. Anda hanya perlu menulis kod-kod arahan yang diperlukan ke dalam buku jawapan supaya atur cara tersebut dapat dilaksanakan mengikut spesifikasi atur cara yang diberikan di atas)

```
/* Atur cara untuk mengira gaji kasar seminggu bagi pekerja-pekerja di
Syarikat XYZ*/
```

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
int no_pekerja, jum_jam;
```

```
float kadar_gaji, gaji_kasar;
```

```
char ulang;
```

```
/* Tuliskan protaip fungsi bagi 2 fungsi yang akan dipanggil oleh fungsi
main*/
```

```
(i)
```

```
ulang = 'Y';
```

```
while (ulang == 'Y')
```

```
{
```

```
printf("\nSila masukkan nombor pekerja : ");
```

```
scanf("%d",&no_pekerja);
```

```
printf("\nSila masukkan kadar gaji sejam : ");
```

```
scanf("%f",&kadar_gaji);
```

```
printf("\nSila masukkan jumlah jam bekerja seminggu : ");
```

```
scanf("%d",&jum_jam);
```

```
/*Tuliskan struktur pilihan untuk memanggil 2 fungsi bergantung
kepada jumlah jam. */
```

```
(ii)
```

```
printf("\n%d mendapat gaji kasar sebanyak
%5.2f",no_pekerja,gaji_kasar);
```

```
printf("\nAdakah anda ingin mengira gaji pekerja lain (Y/T)? : ");
```

```
scanf("%c",&ulang);
```

```
/*akhir while*/
```

```
/*akhir main*/
```

```
/*Sambungan atur cara di muka sebelah*/
```

/*Tuliskan definisi fungsi untuk mengira gaji kasar bagi jumlah jam ≤ 40 */

(iii)

/*Tuliskan definisi fungsi untuk mengira gaji kasar bagi jumlah jam > 40 */

(iv)

[35/100]

- oooOooo -